**Godavari College Of Engineering, Jalgaon.**

**Subject Name:** PDE **Teacher Name:** Prof.Nilesh Chaudhari Sir.

**Practical No**. : 5 **Date:**

**Class: S**.E **Roll No: 1951711245011**

**Title:**  Explain E-R Diagram in Software Engineering..

## **ER diagram :-** An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

ER diagrams are related to data structure diagrams (DSDs), which focus on the relationships of elements within entities instead of relationships between entities themselves. ER diagrams also are often used in conjunction with data flow diagrams (DFDs), which map out the flow of information for processes or systems.

## **When to draw ER Diagrams ?**

So, when do we draw ERDs? While ER models are mostly developed for designing relational databases in terms of concept visualization and in terms of physical database design, there are still other situations when ER diagrams can help. Here are some typical use cases.

* Database design - Depending on the scale of change, it can be risky to alter a database structure directly in a DBMS. To avoid ruining the data in a production database, it is important to plan out the changes carefully. ERD is a tool that helps. By drawing ER diagrams to visualize database design ideas, you have a chance to identify the mistakes and design flaws, and to make corrections before executing the changes in the database.
* Database debugging - To debug database issues can be challenging, especially when the database contains many tables, which require writing complex SQL in getting the information you need. By visualizing a database schema with an ERD, you have a full picture of the entire database schema. You can easily locate entities, view their attributes and identify the relationships they have with others. All these allow you to analyze an existing database and to reveal database problems easier.
* Database creation and patching - Visual Paradigm, an ERD tool, supports a database generation tool that can automate the database creation and patching process by means of ER diagrams. So, with this ER Diagram tool, your ER design is no longer just a static diagram but a mirror that reflects truly the physical database structure.
* Aid in requirements gathering - Determine the requirements of an information system by drawing a conceptual ERD that depicts the high-level business objects of the system. Such an initial model can also be evolved into a physical database model that aids the creation of a relational database, or aids in the creation of process maps and data flow modes.

## ERD notations guide

An ER Diagram contains entities, attributes, and relationships. In this section, we will go through the ERD symbols in detail.

### Entity

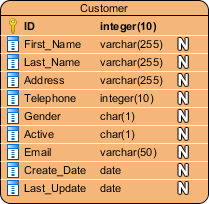
An ERD entity is a definable thing or concept within a system, such as a person/role (e.g. Student), object (e.g. Invoice), concept (e.g. Profile) or event (e.g. Transaction) (note: In ERD, the term "entity" is often used instead of "table", but they are the same). The ERD example below shows an example of an ER entity.



### Entity Attributes

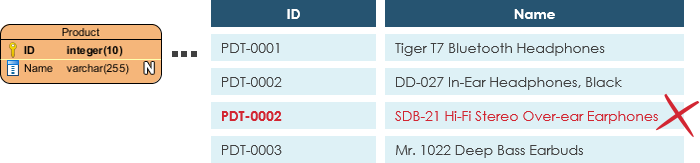
Also known as a column, an attribute is a property or characteristic of the entity that holds it. An attribute has a name that describes the property and a type that describes the kind of attribute it is, such as varchar for a string, and int for integer. When an ERD is drawn for physical database development, it is important to ensure the use of types that are supported by the target RDBMS.

The ER diagram example below shows an entity with some attributes in it.



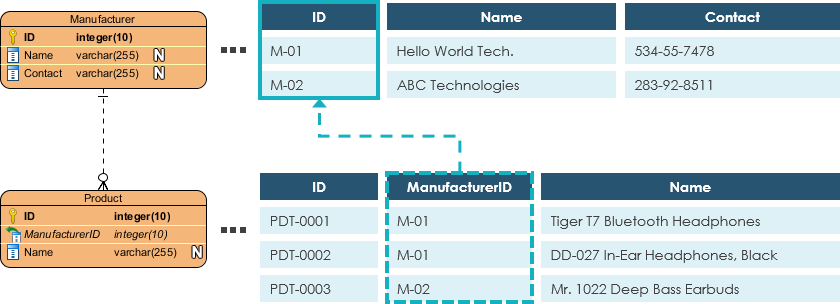
#### Primary Key

Also known as PK, a primary key is a special kind of entity attribute that uniquely defines a record in a database table. In other words, there must not be two (or more) records that share the same value for the primary key attribute. The ERD example below shows an entity 'Product' with a primary key attribute 'ID', and a preview of table records in the database. The third record is invalid because the value of ID 'PDT-0002' is already used by another record.



#### Foreign Key

Also known as FK, a foreign key is a reference to a primary key in a table. It is used to identify the relationships between entities. Note that foreign keys need not be unique. Multiple records can share the same values. The ER Diagram example below shows an entity with some columns, among which a foreign key is used in referencing another entity.



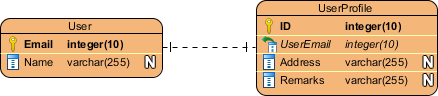
### Relationship :-

A relationship between two entities signifies that the two entities are associated with each other somehow. For example, a student might enroll in a course. The entity Student is therefore related to Course, and a relationship is presented as a connector connecting between them.

#### Cardinality :- Cardinality defines the possible number of occurrences in one entity which is associated with the number of occurrences in another. For example, ONE team has MANY players. The three common cardinal relationships are one-to-one, one-to-many, and many-to-many.

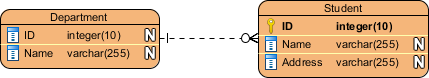
##### One-to-One cardinality example :-

A one-to-one relationship is mostly used to split an entity in two to provide information concisely and make it more understandable. The figure below shows an example of a one-to-one relationship.



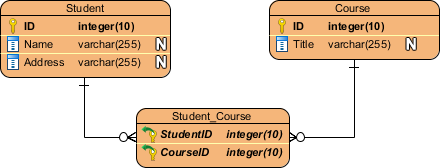
##### One-to-Many cardinality example

A one-to-many relationship refers to the relationship between two entities X and Y in which an instance of X may be linked to many instances of Y, but an instance of Y is linked to only one instance of X. The figure below shows an example of a one-to-many relationship.

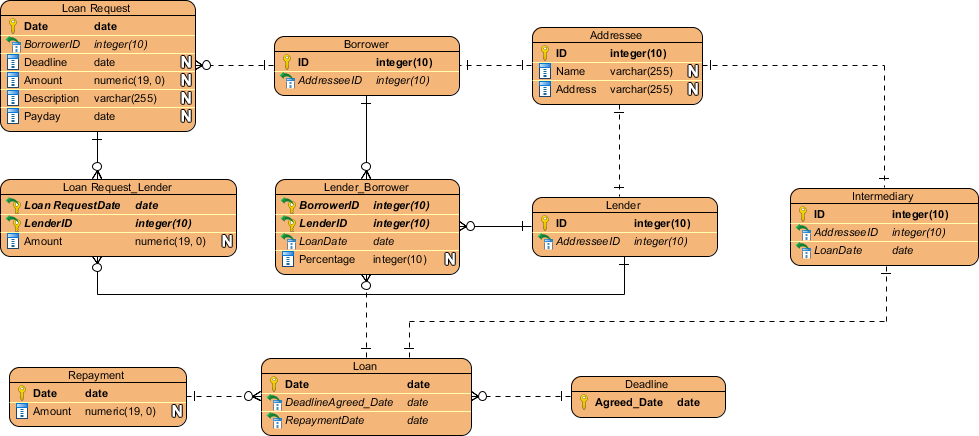


##### Many-to-Many cardinality example

A many-to-many relationship refers to the relationship between two entities X and Y in which X may be linked to many instances of Y and vice versa. The figure below shows an example of a many-to-many relationship. Note that a many-to-many relationship is split into a pair of one-to-many relationships in a physical ERD. You will know what a physical ERD is in the next section.



### **ERD example - Loan System**



**Conclusion:-** I learn from Assignment what is E-R Diagram and how to implement it in Software

Engineeering.